



DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD

GrowRange: Anytime VCG-Based Mechanisms

The Harvard community has made this article openly available.
[Please share](#) how this access benefits you. Your story matters.

Citation	Parkes, David C. and Grant Schoenebeck. 2004. GrowRange: Anytime VCG-based mechanisms. In Proceedings of the 19th national conference on artificial intelligence(AAAI-04): Sixteenth innovative applications of artificial intelligence conference (IAAI-04), ed. American Association for Artificial Intelligence, 34-41. Menlo Park, C.A.: AAAI Press; Cambridge, M.A.: MIT Press.
Published Version	http://www.aaai.org/Papers/AAAI/2004/AAAI04-006.pdf
Accessed	February 18, 2015 3:02:44 PM EST
Citable Link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:4039784
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

(Article begins on next page)

GROWRANGE: Anytime VCG-Based Mechanisms

David C. Parkes

Division of Engineering and Applied Sciences,
Harvard University,
33 Oxford Street, Cambridge MA 02138
parkes@eecs.harvard.edu

Grant Schoenebeck

Harvard College,
Cambridge, MA 02138
schoeneb@fas.harvard.edu

Abstract

We introduce anytime mechanisms for distributed optimization with self-interested agents. Anytime mechanisms retain good incentive properties even when interrupted before the optimal solution is computed, and provide better quality solutions when given additional time. Anytime mechanisms can solve easy instances of a hard problem quickly and optimally, while providing approximate solutions on very hard instances. In a particular instantiation, GROWRANGE, we successively expand the range of outcomes considered, computing the optimal solution for each range. Truth-revelation remains a dominant strategy equilibrium with a stage-based interruption, and is a best-response with high probability when the interruption is time-based.

Introduction

Designing mechanisms to solve distributed optimization problems with self-interested agents is becoming increasingly important in a wide variety of settings, from e-commerce, to the allocation of computational resources in open systems, to planning in multi-agent systems. This field, called *computational mechanism design* (CMD), aims to design solutions that are both incentive-compatible (with truth revelation in a game-theoretic equilibrium) and tractable.

Combinatorial auctions (CAs), with agents that demand bundles of items, are a canonical problem in CMD. All previous work on tractable and strategyproof mechanisms (with truth-revelation in a *dominant-strategy equilibrium*) for CAs has considered restricted domains of agent preferences. For instance, Lehmann et al. (2002) describe a fast and strategyproof CA for single-minded agents that demand only one bundle. But, there are many other examples (Mu'alem & Nisan 2002; Archer et al. 2003; Bartal, Gonen, & Nisan 2003). These methods do not apply to the general CA problem.

In this paper, we introduce *anytime mechanisms*, as a new paradigm for the design of incentive-compatible and tractable mechanisms. Anytime mechanism will solve easy instances of a hard family of problems quickly and optimally, while returning approximate solutions and retaining strategyproofness on very hard instances. Provable worst-case approximation results are dropped in favor of good performance on most problems coupled with the ability to terminate the algorithm with an approximate solution on the

very hardest of problems.¹ We address the challenge of retaining useful incentive properties, such that truthful bidding is an equilibrium whenever a mechanism is terminated.

It is worth emphasizing that a naive approach, in which a Vickrey-Clarke-Groves (VCG) mechanism (see Nisan & Ronen (2000)) is coupled with an anytime winner-determination algorithm, would not be strategyproof unless the algorithm has enough time to solve the problem optimally. The strategyproofness of the VCG mechanism ordinarily relies on the optimality of its decision.

Our solution builds on *maximal-in-range* VCG mechanisms (Nisan & Ronen 2000). We implement anytime mechanisms as *staged* mechanisms, with a new range of outcomes considered in each stage and the optimal solution computed for each new range. Importantly, the range adopted in each stage must not depend on any information reported by agents. In each stage we also compute the optimal solution to the problem without each agent. When interrupted, the anytime mechanism implements the best solution found so far, and determines VCG-based payments on the basis of the best solutions found so far without each agent. The mechanism will continue to compute a better solution when provided with more time.

To understand the incentive properties we consider two different models of interruption. First, we consider stage-based interruptions, in which the mechanism is interrupted after some number of stages. Truth-revelation remains a dominant strategy equilibrium, because the final solution is optimal over the union of the ranges explored, with individual ranges chosen without regard to agent bids. Throughout the paper it is important that the interruption comes from the center, or some third-party, and not from one of the agents.

A more realistic model is one in which the interruption process is *time-based*, for instance an answer might be required after 10 minutes. A new concern here is that an agent can indirectly affect the sequence of ranges explored by changing the difficulty of the problem through its bids, and thus influence the progress made by the algorithm before interruption. Our solution is to use *consensus functions* (Goldberg & Hartline 2003) to compute a conservative and agent-independent estimate of the number of stages completed by the time of an interruption, with results from any additional stages discarded. Together with additional assump-

¹Indeed, Nisan & Ronen (2000) and Lavi et al. (2003) suggest that *no* worst-case polynomial time combinatorial auction can be strategyproof and provide good worst-case approximation properties, without assuming a restricted preference domain.

tions about the maximal influence that an agent can have on the run time, this makes truthful bidding a best-response with high probability, whatever the bids of other agents.

We illustrate our methods in the context of CAs. We define GROWRANGE, which is a particular partition-based instantiation. The empirical results illustrate encouraging performance on hard problems with high run time variance.

Preliminaries

Mechanism design (MD) considers a system of rational self-interested agents and the problem of choosing an outcome k from a finite set K of possibilities. Let N denote the number of agents. Each agent $i \in \mathcal{I}$ has a *type*, $\theta_i \in \Theta_i$, that defines its value $v_i(k, \theta_i)$ for each possible choice k . This information is privately known to each agent. Let $\theta = (\theta_1, \dots, \theta_N)$ denote a type vector and θ_{-i} to denote all types except that of agent i . Agents have *quasilinear* utility, $u_i(k, p) = v_i(k, \theta_i) - p_i$, in value and payments.

Agents report types (perhaps untruthfully) and the mechanism computes the outcome and payments. The challenge is to implement an outcome with good properties despite the ability to misreport types. For instance, *efficient* MD seeks to implement a choice $k_{\text{eff}}(\theta) \in K$ that maximizes the total value across all agents.

Given reports $\hat{\theta}$, mechanism $M = (k, p)$ defines a choice rule $k(\hat{\theta}) \in K$ and a payment rule $p_i(\hat{\theta}) \geq 0$, that determines the payment made by each agent to the center. Let $V(N) = \max_k \sum_i v_i(k, \theta_i)$ and $V(N_{-i}) = \max_k \sum_{j \neq i} v_j(k, \theta_j)$.

Definition 1 *The Vickrey-Clarke-Groves (VCG) mechanism defines choice rule $k_i(\hat{\theta}) = \arg \max_{k \in K} \sum_i v_i(k, \hat{\theta}_i)$, and payment rule $p_i(\hat{\theta}) = v_i(k(\hat{\theta}), \hat{\theta}_i) - [V(N) - V(N_{-i})]$.*

The VCG mechanism has the following properties:

Strategyproof: Truth-revelation is a dominant-strategy equilibrium. Formally, $v_i(k_{\text{eff}}(\theta_i, \theta_{-i}), \theta_i) - p_i(\theta_i, \theta_{-i}) \geq v_i(k_{\text{eff}}(\hat{\theta}_i, \theta_{-i}), \theta_i) - p_i(\hat{\theta}_i, \theta_{-i})$ for all i , all θ_{-i} , all θ_i , and all $\hat{\theta}_i \neq \theta_i$.

Efficient: The choice implemented in the VCG mechanism maximizes the total value across agents, in equilibrium.

Strategyproofness is a useful property for mechanisms because it simplifies the strategic problem facing bidding agents. An agent does not need to model the values or strategies of other agents to compute its equilibrium strategy.

Often times the optimization problem $k(\hat{\theta})$ is intractable, for instance in CAs (Rothkopf, Pekeč, & Harstad 1998). It is interesting to consider a *VCG-based* mechanism, in which the optimal choice rule, k^* , is replaced with an approximate choice rule. Everything else is left unchanged, with payments computed by applying the approximate choice rule to solve the optimization problem without each agent.

In particular, Nisan & Ronen (2000) define *maximal-in-range* VCG mechanisms. Consider some range $\tilde{K} \subseteq K$, and let $V(N; \tilde{K}) = \max_{k \in \tilde{K}} \sum_i v_i(k, \hat{\theta}_i)$ and $V(N_{-i}; \tilde{K}) = \max_{k \in \tilde{K}} \sum_{j \neq i} v_j(k, \hat{\theta}_j)$.

Definition 2 *A maximal-in-range VCG mechanism implements choice rule $\tilde{k}(\hat{\theta}) = \arg \max_{k \in \tilde{K}} \sum_i v_i(k, \hat{\theta}_i)$, and payment rule $p_i(\hat{\theta}) = v_i(\tilde{k}(\hat{\theta}), \hat{\theta}_i) - [V(N; \tilde{K}) - V(N_{-i}; \tilde{K})]$, for some range $\tilde{K} \subseteq K$.*

Proposition 1 (Nisan & Ronen 2000) *A VCG-based mechanism is strategyproof if and only if it is maximal-in-range.*

That maximal-in-range is sufficient for strategyproof follows immediately from the strategyproofness of VCG mechanisms. Maximal-in-range is also necessary for strategyproofness because otherwise there is always a set of types for which one agent can select the maximal choice in the range by misreporting its type, and thus improve its utility.

Crucially, the strategyproofness of maximal-in-range approximations hinges on the agents retaining an *expressive language* for outcomes in the range, and on the range being selected *independently* of agent bids.

Combinatorial Auctions

Combinatorial auctions (CAs) have received particular attention in CMD. In a CA, there is a set of \mathcal{G} items to allocate. Agent i with type θ_i has value $v_i(S, \theta_i) \geq 0$ for a bundle $S \subseteq \mathcal{G}$ of items. For simplicity, it is convenient to drop the explicit dependence on type and write $v_i(S)$. We assume *free-disposal*, with $v_i(T) \geq v_i(S)$ for all $T \supseteq S$ and normalize with $v_i(\emptyset) = 0$. The efficient allocation maximizes $\sum_i v_i(S_i)$ subject to $S_i \cap S_j = \emptyset$ and $S_i \subseteq \mathcal{G}$.

Noting that the strategyproofness of the maximal-in-range approximation requires no restrictions on agent valuations, one might wonder whether a maximal-in-range approximation can provide a CA that is tractable and produce reasonable solutions. Unfortunately, the answer is negative.

Theorem 1 (Nisan & Ronen 2000) *No strategyproof VCG-based CA can be both tractable and reasonable.*

In particular, an allocation is said to be *reasonable* if whenever a single agent values an item that agent receives the item in the allocation. Without reasonableness there can be no useful worst-case guarantee on efficiency. Thus, this negative result makes the case for an anytime approach. Without this, we must either: (a) impose an *a priori* restriction on the range that will sometimes preclude reasonable behavior even when an instance was actually easy; or (b) seek optimality and accept that sometimes a solution will not be returned quickly; or (c) accept a loss in strategyproofness and an unraveling of incentives.

We note, parenthetically, that although there are many known tractable special-cases of the CAP that arise due to restrictions on the structure of bundles (e.g. *circular-ones*, *consecutive-ones*, *two-ones*) (Rothkopf, Pekeč, & Harstad 1998; de Vries & Vohra 2003), and that thus suggest maximal-in-range approximations, all of them assume an *additive-or* language for bids. This language is *not* expressive for general valuations, even on restricted ranges (unless the range only allows each agent to receive a single bundle). Moreover, introducing an expressive language, with *exclusive-or* bids (Nisan 2000), requires introducing additional side constraints that break the very structure that is required for tractability.

Anytime VCG-Based Mechanisms

We use maximal-in-range VCG approximations to define anytime VCG mechanisms. The mechanisms are defined for a sequence of ranges K^1, K^2, \dots , that satisfy the following properties:

1. The sequence of ranges is independent of reported types.
2. Optimization $V(N; K^z)$ and $V(N_{-i}; K^z)$ on the range K^z in early stages z tends to be easier than solving the complete problem $V(N; K)$.
3. The union of ranges, $\bigcup_z K^z = K$, i.e. the search eventually computes the optimal solution across all choices.

It is critical that the sequence of ranges not depend on the bids from agents to maintain strategyproofness. As an example, this rules out allowing agents to submit 10 bids of their choice, then 20 bids, then 30 bids, \dots with the range defined in terms of the bundles in agent bids.²

In our staged, maximal-in-range, approach we consider a sequence of ranges, K^1, K^2, \dots . Let $K^{<z} = \bigcup_{z' < z} K^{z'}$, and $K^{\leq z} = \bigcup_{z' \leq z} K^{z'}$. Let \tilde{k}^z denote the solution to $\max_{k \in K^z} \sum_i v_i(k, \hat{\theta}_i)$, and \tilde{k}_{-i}^z denote the solution to $\max_{k \in K^z} \sum_{j \neq i} v_j(k, \hat{\theta}_j)$.

Definition 3 (Anytime VCG Mechanism) Given a sequence of ranges K^1, K^2, \dots and reports $\hat{\theta}$:

1. In stage z , solve $\max_{k \in K^z} \sum_i v_i(k, \hat{\theta}_i)$ and $\max_{k \in K^z} \sum_{j \neq i} v_j(k, \hat{\theta}_j)$ for all $i \in N$. Update \tilde{k}^z and \tilde{k}_{-i}^z , to maintain the best solutions seen so far.
2. If an interruption is received during the stage z' then implement outcome $\tilde{k}^{z'-1}$, and payments
$$p_i = v_i(\tilde{k}^{z'-1}, \hat{\theta}_i) - [V(N; K^{<z'}) - V(N_{-i}; K^{<z'})].$$

Note that the payments can be quickly computed by evaluating the best solutions determined at the end of stage $z' - 1$; it is not necessary to solve any additional optimization problems. Also, note that this is not simply a sequence of VCG-based mechanisms, because the VCG payments are computed in terms of the best solutions across all stages, regardless of the stage in which they occurred. Not all best solutions (for the main problem and the problem without each agent) need occur in the same stage.

Turning to incentives, consider a stage-based interruption, for instance an answer might be required after 10 stages. More generally, the interruption can follow some p.d.f. $c(z)$, that can be known to the agents.

Theorem 2 *The anytime VCG mechanism is strategyproof for a stage-based interruption $c(z)$, and will implement the efficient allocation if allowed to run for enough stages.*

Proof: The anytime VCG mechanism is strategyproof for any fixed number of stages because the sequence of ranges are independent of agent bids. Thus, the union over some number of ranges defined by an agent-independent distribution is itself agent independent. Efficiency holds once the final stage is implemented because this range contains all feasible solutions. ■

²This scheme was proposed in Banks et al. (1989).

Avoiding Redundant Computation

A number of simple optimizations are possible to avoid the unnecessary duplication of computation across stages.

We describe these in the context of an implementation in which the optimization problems in each stage are formulated as *mixed-integer programs*, and solved via branch-and-cut LP-based search. But the ideas are general, and also hold for other systematic search algorithms.

Lower-bound Pruning. When solving $V(N; K^z)$ and $V(N_{-i}; K^z)$ in stage z , use the value of the best current solutions, i.e. $V(N; K^{<z})$ and $V(N_{-i}; K^{<z})$ to prune the search. For instance, when all solutions under a search node are dominated in value by the current best solution then prune the subtree at that node.

Second-best Solutions. When solving $V(N_{-i}; K^z)$ for stage z , we know $V(N_{-i}; K^z) \leq V(N; K^z)$. Solve for $V(N; K^z)$ first, and then if a feasible solution to $V(N_{-i}; K^z)$ also has value $V(N; K^z)$ then immediately stop without *proving* optimality. Combining with lower-bound pruning, if $V(N_{-i}; K^{<z}) \geq V(N; K^z)$ then there is no need to solve $V(N_{-i}; K^z)$.

Disjunctive Search. Constraints can be added to the formulation of search in a new stage to avoid duplication of effort with earlier searches. For instance, the problem $V(N; K^z)$ can be formulated as $\max_{k \in K^z} \sum_i v_i(k, \hat{\theta}_i)$ s.t. $k \notin K^{<z}$. This is related to the idea of “local branching” in Fischetti & Lodi (2002). In the case of CAs, this means adding a constraint to ensure that at least one agent receives a new bundle.

An Anytime VCG-Based Mechanism for CAs

GROWRANGE is a concrete instantiation of an anytime VCG mechanism for CAs. In particular, it adopts a *partition-based* sequence of ranges, and considers only monotonically-increasing sequences of ranges.

Partition-Based CAs

Let $\Pi = \{A_1, \dots, A_k\}$ denote a partition of items \mathcal{G} into non-empty parts. Following Holzman et al. (2001), let $\Sigma_\Pi \subseteq 2^{\mathcal{G}}$ denote the set of bundles generated by Π , i.e. containing all bundles of the form $\bigcup_{l \in L} A_l$, where $L \subseteq \{1, \dots, k\}$. Clearly, the “grand” bundle \mathcal{G} is always in Σ_Π , and we also require that Σ_Π includes the empty bundle, \emptyset . Given all bundles in Σ_Π we define the *range* to include all allocations $S = (S_1, \dots, S_N)$ in which $S_i \in \Sigma_\Pi$ for every agent i . Loosely, we also find it convenient to simply refer to the bundles, Σ_Π , as the range.

This leads to a concrete instantiation of a maximal-in-range CA. Consider an exclusive-or (XOR) bidding language (Nisan 2000), with bid (b_i, B_i) from agent i defining value $\hat{v}_i(S)$ on all $S \in B_i$ and value $\hat{v}_i(S) = \max_{S' \in B_i, S' \subseteq S} b_i(S')$ on all bundles $S \subseteq \mathcal{G}$, $S \notin B_i$, by free-disposal. Projecting these bids onto the restricted range, Σ_Π , we construct bids $b'_i(S) = \hat{v}_i(S)$ on all bundles $S \in B'_i = \bigcup_{S \in B_i} C(S)$ with $C(S) \in \Sigma_\Pi$ defined as the bundle in Σ_Π that minimally includes S (Holzman et al. 2001). There is a unique such bundle because the field,

Σ_{Π} , is closed under intersection. The premise behind this approach is that fewer bids tend to make winner determination easier. Note that the input size of the projected bids from agent i is at most $\min\{|B_i|, |\Sigma_{\Pi}|\}$.

Given these projected bids, a standard “branch-on-bid” winner-determination algorithm such as ILOG’s CPLEX mixed-integer programming package, or special-purpose solvers such as CABOB (Sandholm *et al.* 2001) can be used to compute the optimal solution and provide a maximal-in-range VCG mechanism.

As an example, consider an agent with valuation $\{(a, \$50), (b, \$100)\}$, goods $\mathcal{G} = \{a, b, c, d\}$, and partitions $\Pi^1 = \{abc, d\}$, $\Pi^2 = \{a, bc, d\}$, $\Pi^3 = \{a, b, c, d\}$, that define fields $\Sigma_{\Pi^1} = \{\emptyset, abc, d, abcd\}$, $\Sigma_{\Pi^2} = \{\emptyset, a, ad, abc, bc, bcd, d\}$ and $\Sigma_{\Pi^3} = 2^{\mathcal{G}}$. The projected bids are $\{(abc, \$100)\}$, $\{(a, \$50), (bc, \$100)\}$, and $\{(a, \$50), (b, \$100)\}$, for partitions Π^1 , Π^2 , and Π^3 respectively.

The additional structure in a partition-based range can also be made available to a solver by defining a new set of “dummy” items, each associated with one bundle in the partition. In fact, the special structure of a partition-based range provides an additional equilibrium property, defined with respect to the bids submitted on the range Σ_{Π} :

Theorem 3 (Holzman *et al.* 2001) *No agent can benefit by unilaterally submitting a bid outside of a partition-based range Σ_{Π} , given that bids from other agents are projected onto Σ_{Π} .*

Algorithm: GROWRANGE(k_0, α)

A sequence of partitions, Π^1, Π^2, \dots , defines the sequence of ranges. Parameter k_0 defines the size of the initial partition, and parameter α defines the number of refinements made in between stages:

- To construct Π^0 , choose k_0 items uniformly at random, to seed a separate component of Π^0 . Place each remaining item uniformly at random into one of the components.
- To construct Π^z from Π^{z-1} , choose a component $S \in \Pi^{z-1}$ with $|S| > 1$ and split the component (choosing the split uniformly at random) into two new components. Repeat α times (or until all components of the partition are singletons).

Note that because we define refinements of the partition by this subdivision process each successive range will be defined on more and more bundles, *i.e.* $\Sigma_{\Pi^z} \supseteq \Sigma_{\Pi^{z-1}}$. This simplifies the anytime VCG mechanism, because the best solutions to $V(N; K^{\leq z})$ and $V(N_{-i}; K^{\leq z})$, for any z , will always be those computed in the most recent stage. Also, we see that the solutions used to determine payments are determined in the *same* stage as the best overall solution.

Empirical Analysis

In preliminary studies, we have investigated the anytime performance of GROWRANGE for CAs. For winner determination, we use the IBM OSL mixed-integer programming (MIP) solver both to solve optimization problems within

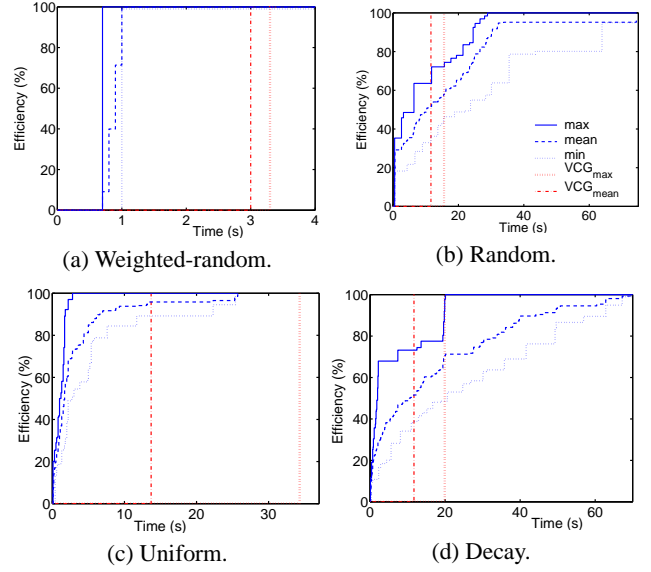


Figure 1: Anytime performance of GROWRANGE.

each stage of GROWRANGE, as well as to compute the outcome for the one-shot (optimal) VCG mechanism.

In testing GROWRANGE, we adapt the bid distributions defined in Sandholm (1999), but using Sandholm’s models to define *valuations* for agents. It is important to note that we generate values with these distributions, and not bid prices. In particular, we adopt an *exclusive-or* (XOR) logic to define a valuation function in terms of the values on a sparse number of bundles.³ We adapt the *Weighted-random*, *Random*, *Uniform* and *Decay* distributions. Problem sizes are (agents/bundles-per-agent/goods): 100/4/200, 80/4/160, 40/4/80, and 80/4/160 for each distribution respectively, with Sandholm’s Decay parameter set to 0.55 and with bundles of size 5 in Uniform. We adopt parameters $(k_0, \alpha) = (5, 10)$ for all distributions except Uniform, for which we adopt parameters $(k_0, \alpha) = (2, 5)$.

Figure 1 illustrates the performance of GROWRANGE. We plot the average and maximum runtime of the one-shot VCG mechanism, along with the *max*, *min*, and *average* anytime profiles of GROWRANGE. All results are averaged over 10 instances, and experiments were performed on a 4 GHz P4, with 512 MB RAM. Note that we measure the total run time for GROWRANGE, including the overhead for projecting bids and constructing partitions.

Given that we use *random* partitions in GROWRANGE and make no effort to tune the sequence of partitions to the prob-

³It is a common misconception that these distributions provide easy winner-determination problems. As initially defined in Sandholm (1999) the distributions were used to define bids connected with an *additive-or* bidding language. It is in this form that Andersson *et al.* (2000), and others, have shown that the Weighted Random, Random, and to some extent the Decay distributions are easy. We are not aware of any studies of the complexity of winner determination when these distributions are used to generate values, and structured as an XOR’ed set.

lem domain we find these initial results quite encouraging. If a system was to use the one-shot VCG mechanism it would need to be prepared to wait for the *maximal* possible run time, because if there was ever a chance that it would be terminated before completion then strategyproofness would be lost and the performance would unravel. Thus, the most interesting test for GROWRANGE is to look at its anytime performance after some fraction of the maximal VCG run time.

The anytime approach is most promising on problems for which there is a large variance in solution difficulty across instances. This is illustrated on Uniform, for which the run time of the VCG mechanism has a large variance (the mean run time is 13.7s but the maximal run time was 34.3s). GROWRANGE averages better than 90% allocative-efficiency in 50% of the worst-case VCG run time. We also see good performance on Weighted-random, although this proved to be an easy problem for both methods.

The *Decay* and *Random* distributions proved not to be well suited to the anytime approach because we found little variance in the run time of the VCG mechanism, at least over 10 trials. GROWRANGE averages only around 50% efficiency in 50% of the worst-case VCG run time. Future analysis should take additional samples to make sure that there is indeed little variance of VCG run time on these problems.

Leyton-Brown et al. (2002) have studied the variance in run time for CAs, given distributions from CATS (?). Their analysis suggests a high variance in many problems. We view this as further evidence that anytime mechanisms are necessary in practice.⁴

Time-Based Interruptions

We now turn to a time-based model of interruption, in which the center interrupts the mechanism according to some random process $T \sim c(t)$ that defines the time T at which a solution is required. Time-based interruptions are appealing because they allow the center to respond when challenged to provide a solution, and also because they can facilitate the integration of cost-of-delay based models of meta-deliberation (Horvitz 1987; Dean & Boddy 1988).

Even when the sequence of ranges K^1, K^2, \dots remains independent of bids from agents, an agent can try to change the *progress* made across this sequence by a particular time by submitting bids that change the difficulty of the optimization problems in each stage. Consider, for instance, an agent that knows that it is a winner in early stages but not in later stages. If an interrupt will come quite late, this agent could try to state a type that *slows* down computation in early stages. Of course, this is not without drawbacks because the agent must also be careful not to adversely change the outcome in any stages that are searched (it's dominant strategy, contingent on a fixed or random range, remains truth-revelation).

A simple fix would use an optimization algorithm with a run time that is the same for all bids, for any given range.

⁴Subject to the concern, discussed in the next section, that this variance *not* be accompanied by significant opportunities for any *single* agent to change the run time through its bid.

But, this approach is not appealing because we want to solve easy instances quickly rather than design for the worst-case. This is the whole motivation for an anytime approach!

Instead, we define a randomized mechanism and retain strategyproofness with some error probability (Archer *et al.* 2003). We say a mechanism M is *strategyproof with error probability p* if for every θ_{-i} and every θ_i , the probability that a truthful bid θ_i is not the best-response for agent i is no greater than p .

Let $z(\theta, t)$ denote the number of stages completed by time t . We use *consensus functions* (Goldberg & Hartline 2003) to compute a conservative estimate $f(\theta, t) \leq z(\theta, t)$ of the number of stages, that is agent independent and cannot be manipulated by any single agent with probability p . The mechanism then “rolls-back” its state to the end of stage $f(\theta, t)$, discarding any results from additional computation.

For now, we provide a high-level description of the mechanism. Let $time(\theta, z)$ denote the total compute time to solve $V(N; K^{\leq z})$ and $time(\theta_{-i}, z)$ denote the total compute time to solve $V(N_{-i}; K^{\leq z})$, given reports θ . We define and construct our agent-independent estimator below.

Definition 4 (Consensus-Based Anytime VCG Mech.)

Given a sequence of ranges K^1, K^2, \dots , reports $\hat{\theta}$, and a conservative and agent-independent estimator $f(\theta, t)$:

1. In stage z , solve $\max_{k \in K^z} \sum_i v_i(k, \hat{\theta}_i)$ and $\max_{k \in K^z} \sum_{j \neq i} v_j(k, \hat{\theta}_j)$ for all $i \in N$. Update \tilde{k}^z and \tilde{k}_{-i}^z , to maintain the best solutions seen so far. Update $time(\hat{\theta}, z)$ and $time(\hat{\theta}_{-i}, z)$.
2. If an interruption is received at time t' , then compute $z' = f(\hat{\theta}, t')$, and implement outcome $\tilde{k}^{z'ce}$, and payments

$$p_i = v_i(\tilde{k}^{z'}, \hat{\theta}_i) - [V(N; K^{\leq z'}) - V(N_{-i}; K^{\leq z'})].$$

Discard the results from additional computation completed after stage z' .

Theorem 4 *Consensus-Based Anytime VCG Mechanism, with conservative estimator $f(\theta, t)$ that is agent independent with probability p , and time-based interruption $c(t)$, is strategyproof with probability p .*

Proof: Whatever the reports θ_{-i} of other agents $\neq i$, and whatever the interruption time t' , with probability p the mechanism is rolled-back to the same stage $z' = f(\hat{\theta}_i, \theta_{-i}, t')$ for all reports $\hat{\theta}_i$. Agent i 's best-strategy is to bid truthfully when this occurs, because the mechanism will implement a maximal-in-range VCG outcome on range $\bigcup_{z \leq z'} K^z$. ■

Here, it is important that the second-best solutions $V(N_{-i}; K^z)$ are *explicitly* solved in each stage, in order to construct a suitable agent-independent estimator $f(\theta, t)$.⁵ Also, it is important that the estimate is close to the true

⁵In comparison, these calculations can be skipped in the stage-based interruption model when the solution to $V(N_{-i}; K^z)$ can be readily inferred from $V(N; K^z)$. For instance, in a CA this occurs when agent i receives no items in the efficient solution.

number of stages completed because we must discard any computation performed after the estimated stage $f(\theta, t')$ given interruption at time t' .

An Agent-Independent Stage Estimator

We need an estimator $f(\theta, t)$ for the number of completed stages that is agent independent with high probability.

Definition 5 Function $f(\theta, t) \in \mathbb{N}$ is a *conservative and agent-independent estimator with probability p* , if:

- P1. $f(\theta, t) \leq z(\theta, t)$, for all θ and all t .
- P2. $f(\hat{\theta}_i, \theta_{-i}, t) = f(\theta_i, \theta_{-i}, t)$, for any θ_{-i}, θ_i , and t , and all $\hat{\theta}_i \neq \theta_i$, with probability p over coin flips independent of θ and t .

A trivial estimator $f(\theta, t)$ would return 0 for small times t and then 1 for larger times, for all θ . This would be conservative and agent independent, but not informative. We will seek an estimator that returns a number of stages $f(\theta, t)$ “close” to the actual number of completed stages, $z(\theta, t)$. Parenthetically, we note that no estimator can be perfectly agent independent and informative:

Lemma 1 If f is agent independent, then f depends only on t .

Proof: Suppose not, then there exist θ, t such that $f(\theta, t) \neq f(\theta', t)$, so we can change θ to θ' an agent at a time to get the chain $\theta = \theta^0, \dots, \theta^n = \theta'$, where θ^k has the first k types from θ' and the rest from θ . Then by the agent independence of f : $f(\theta^k, t) = f(\theta^{k+1}, t)$ and so $f(\theta, t) = f(\theta', t)$, a contradiction. ■

We require two assumptions about the effect that a single agent’s bid can have on the run time.

Assumption 1 (ν -Closeness) No one problem is more than $\nu > 1$ times slower than any other problem across the same sequence of ranges:

$$\frac{\max[\{time(\theta_{-i}, z) : i \in \mathcal{I}\}, time(\theta, z)]}{\min[\{time(\theta_{-i}, z) : i \in \mathcal{I}\}, time(\theta, z)]} \leq \nu, \quad \forall z$$

for any sequence of ranges, K^1, K^2, \dots , and for all types θ .

Assumption 2 (γ -bounded single-agent slow-down) No single agent can slow down the computation to any problem across any sequence of stages by more than a factor $\gamma > 1$ by stating a false report. For any agent i ,

$$\frac{time(\hat{\theta}_i, \theta_{-i}, z)}{time(\theta_{-j}, z)} \leq \gamma, \quad \forall \hat{\theta}_i \neq \theta_i, \forall \theta_{-j}, \forall \theta_i, \forall z$$

$$\frac{time(\hat{\theta}_i, \theta_{-i}, z)}{time(\theta, z)} \leq \gamma, \quad \forall \hat{\theta}_i \neq \theta_i, \forall \theta_{-i}, \forall \theta_i, \forall z$$

for any sequence of ranges, K^1, K^2, \dots .

Ultimately, both assumptions can be empirically validated for a domain in question. The tighter that ν - and γ - are in practice, the better the stage estimate will approximate the

actual number of stages completed when the mechanism is interrupted.⁶

Following Goldberg & Hartline (2003), who use $g_{u,c}$ in a very different context, we now define a random function $g_{u,c}(t)$ that is a “ ρ -consensus estimator” at time t with probability p , for some $\rho > 1$.

Definition 6 (consensus-estimator $g_{u,c}$) Fix some $c > 1$, then define:

$$g_{u,c}(t) = \min_{j \in \mathbb{Z}} c^{u+j} \text{ such that } t \leq c^{u+j} \quad (1)$$

with u chosen uniformly at random from $[0, 1]$.

Parameter c in $g_{u,c}$ can be used to make a trade-off in our mechanism between the probability, p , that truthful bidding is an equilibrium and the amount of roll-back that is required on interruption. We discuss this further at the end of this section.

Definition 7 We say $g : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is a ρ -consensus estimator function at time t if:

- $g(t) \geq t$
- $g(t) = g(t') \forall t' \text{ such that } t/\rho \leq t' \leq t$.

for some $\rho > 1$.

A function g with these properties is named a “consensus estimator” because given a set of estimates t_1, t_2, \dots , then a consensus estimator g at $\max\{t_1, t_2, \dots\}$ would return $g(t_1) = g(t_2) = g^*$ for all estimates whenever the individual estimates t_1, t_2, \dots are clustered within multiplicative factor $\rho > 1$ of each other.

The family, $g_{u,c}$, is defined as a family of *step functions* that are flat on large regions. This ensures that the second condition for a consensus-estimator is often satisfied. The relative size of these flat regions (defined by c) makes a tradeoff between the probability of consensus and the accuracy of the estimate.⁷ The selection of a random u ensures that the function $g_{u,c}$ is equally likely to be a consensus estimator function at any t :

Lemma 2 Randomly chosen $g_{u,c}$ is a ρ -consensus estimator at t with probability $p = 1 - \log_c(\rho)$, for any $\rho > 1$.

Proof: To see this we fix an t and integrate over the possible values for u that give us a ρ -consensus estimation function. Let $\pi(u)$ denote the p.d.f. for u , Uniform on $[0, 1]$. Without loss of generality, assume that $\frac{t}{\rho} = c^j$ for some $j \in \mathbb{Z}$. Then $c^{j+u} \in [\frac{t}{\rho}, \frac{ct}{\rho}]$. Note that $c^{j+u} \in [\frac{t}{\rho}, t)$ iff this is not a ρ -consensus estimation function.

$$\int_{c^{u+j}=\frac{t}{\rho}}^{c^{u+j}=t} \pi(u) du = \int_{u=(\log_c \frac{t}{\rho})-j}^{u=(\log_c t)-j} \pi(u) du$$

$$= (\log_c t - j) - (\log_c \frac{t}{\rho} - j) = \log_c \rho = 1 - p.$$

⁶Although Leyton-Brown et al. (2002) suggest that the empirical hardness of winner-determination in CAs may vary by orders of magnitude for different instances from a distribution, we are not aware of any research that has considered the possible effect that changing the bid from a *single* agent can have on run time.

⁷Goldberg & Hartline (2003) show there can be no function g that works as a consensus estimator for all values t with certainty.

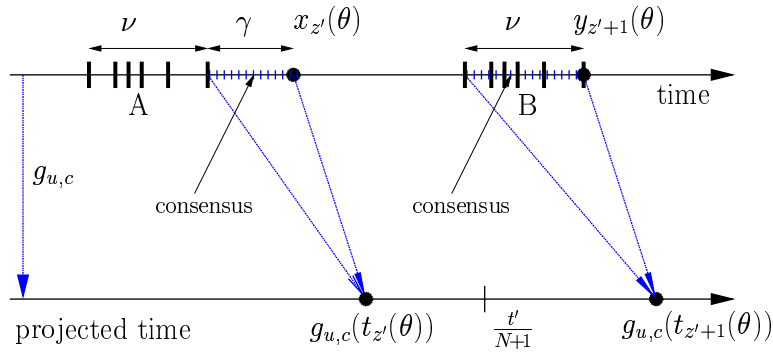


Figure 2: Analysis of Agent-Independent Stage Estimator.

The conservative and agent-independent stage estimator $f(\theta, t)$ is defined in terms of $\text{time}(\hat{\theta}, z)$, $\text{time}(\hat{\theta}_{-i}, z)$, and consensus-estimator $g_{u,c}$:

Definition 8 (agent-independent stage estimator) Fix $c > 1$. Given interrupt t' , and reports $\hat{\theta}$:

1. Select u uniformly at random from $[0, 1]$.
2. Compute $f(\hat{\theta}, t')$ as:

$$\begin{aligned} f(\hat{\theta}, t') &= \max_{z \geq 0} z \\ \text{s.t. } (N+1) \cdot g_{u,c}(\text{time}(\hat{\theta}_{-i}, z)) &\leq t', \quad \forall i \\ (N+1) \cdot g_{u,c}(\text{time}(\hat{\theta}, z)) &\leq t' \end{aligned}$$

where N is the number of agents.

Consider Figure 2. Let $z' = f(\theta, t')$, given reports θ and interrupt t' , and define $t_z(\theta) = \max[\{\text{time}(\theta_{-i}, z) : i \in \mathcal{I}\}, \text{time}(\theta, z)]$, and special points $x_{z'}(\theta)$ and $y_{z'+1}(\theta)$, as follows:

$$\begin{aligned} x_{z'}(\theta) &= \gamma \cdot t_{z'}(\theta) \\ y_{z'+1}(\theta) &= t_{z'+1}(\theta) \end{aligned}$$

Lemma 3 Estimator $f(\theta, t')$ is agent independent when $g_{u,c}$ provides a γ -consensus at $x_{z'}$ and a ν -consensus at $y_{z'+1}$.

Proof: Consider Figure 2. Consider agent i , and suppose the reports from other agents are θ_{-i} . Let $z' = f(\theta, t')$. From the definition of $f(\theta, t)$, we know that:

$$\begin{aligned} g_{u,c}(t_{z'}(\theta)) &\leq \frac{t'}{N+1} \\ \text{and } g_{u,c}(t_{z'+1}(\theta)) &> \frac{t'}{N+1} \end{aligned}$$

The report, $\hat{\theta}_i \neq \theta_i$, from agent i can only change the stage returned by f if: a) one of $\{\text{time}(\hat{\theta}_i, \theta_{-ij}, z') : j \neq i\}$, or $\text{time}(\hat{\theta}_i, \theta_{-i}, z')$ increases so that $g_{u,c}(t_{z'}(\hat{\theta}_i, \theta_{-i})) > t'/(N+1)$; or b) all of $\{\text{time}(\hat{\theta}_i, \theta_{-ij}, z' + 1) : j \neq i\}$, $\text{time}(\theta_{-i}, z' + 1)$, and $\text{time}(\hat{\theta}_i, \theta_{-i}, z' + 1)$ decrease so that $g(t_{z'+1}(\hat{\theta}_i, \theta_{-i})) \leq t'/(N+1)$. But, a) is not possible because we have a γ -consensus at $x_{z'}(\theta)$ and therefore

$g_{u,c}(x_{z'}(\theta)) = g_{u,c}(t_{z'}(\theta)) = g_{u,c}(t)$ for all $x_{z'}(\theta)/\gamma \leq t \leq x_{z'}(\theta)$ and the most agent i can slow down any time in group **A** (see Figure 2) is by a factor of γ because of the γ -bounded single-agent slow-down assumption. Also, b) is not possible because there is one point, $\text{time}(\theta_{-i}, z' + 1)$, that agent i cannot move, and because a ν -consensus at $y_{z'+1}(\theta)$ implies that all points in group **B** (see Figure 2) map to the same point $g_{u,c}(\text{time}(\theta_{-i}, z' + 1))$ to the right of $t'/(N+1)$, and to (at least) $g_{u,c}(\text{time}(\theta_{-i}, z' + 1))$ must remain to the right of $t'/(N+1)$ for any report θ_i . ■

Proposition 2 Estimator $f(\theta, t')$ is a conservative estimator, and agent independent with probability $p = 1 - \log_c \rho$ for $\rho = \nu \cdot \gamma$, for any time t' .

Proof: Let $z(\theta, t')$ denote the number of stages completed by time t' . We know that $z' = f(\theta, t') \leq z(\theta, t')$, given interrupt t' , because

$$\begin{aligned} t' &\geq (N+1)g_{u,c}(\max[\{\text{time}(\theta_{-i}, z') : i \in \mathcal{I}\}, \text{time}(\theta, z')]) \\ &\geq (N+1) \max[\{\text{time}(\theta_{-i}, z') : i \in \mathcal{I}\}, \text{time}(\theta, z')] \\ &\geq \sum_i \text{time}(\theta_{-i}, z') + \text{time}(\theta, z'), \end{aligned}$$

which is the actual run time required to complete z' stages.

For agent independence, the probability that we do *not* have a γ -consensus at $x_{z'}(\theta)$ is $\log_c \gamma$ by Lemma 2, similarly the probability that we do *not* have a ν -consensus at $y_{z'+1}(\theta)$ is $\log_c \nu$. The probability of not having a γ -consensus or not having a ν -consensus is at most $\log_c \gamma + \log_c \nu = \log_c \rho$, where ρ is defined as $\gamma \cdot \nu$. Thus, the probability of agent independence is at least $1 - \log_c \rho$. ■

This leads to our main result.

Theorem 5 The Consensus-Based Anytime VCG Mechanism is strategyproof with probability $1 - \log_c \rho$ for a time-based interruption $c(t)$, and will implement the efficient allocation if allowed to run for enough stages.

Cost of Consensus

What is the role of parameter c ? Well, for larger values of c , probability p gets closer to 1 but with an estimate that tends to be further away from the true number of stages completed. Considering the error introduced by consensus-estimator function $g_{u,c}$, we have:

Lemma 4 The expectation of $g_{u,c}(t)/t$ given that $g_{u,c}$ is a ρ -consensus estimation function at t is $\frac{c-\rho}{(\ln c - \ln \rho)\rho}$.

Proof: Fix t , and assume that $\frac{t}{\rho} = c^j$ for some $j \in \mathbb{Z}$. This is a ρ -consensus estimation function iff $c^{j+u} \in [t, \frac{ct}{\rho}]$. So

$$\int_{c^{u+j}=t}^{c^{u+j}=\frac{ct}{\rho}} \frac{\pi(u)}{t} du = \frac{c^j}{t(1 - \log_c \rho)} \int_{u=(\log_c t)-j}^{u=(\log_c \frac{ct}{\rho})-j} c^u du = \frac{c^j}{t(1 - \log_c \rho)} \left(\frac{c^{\log_c(\frac{ct}{\rho})-j} - c^{\log_c(t)-j}}{\ln c} \right) = \frac{c - \rho}{(\ln c - \ln \rho)\rho}$$

Putting this together, the overhead introduced through the consensus-estimate methods gives a run time for $z' = f(\theta, t')$ usable stages that scales as a multiple:

$$\nu \cdot \frac{c - \gamma}{(\ln c - \ln \gamma)\gamma} \cdot T(z', \theta)$$

of the run time, $T(z', \theta)$, for z' completed stages in the anytime VCG mechanism defined for stage-based interruptions, for the same sequence of ranges. The ν factor accrues because we must take the maximal time across the $time(\theta, z')$ points rather than the average. The second term follows from the error analysis in Lemma 4 and because we have a γ -consensus estimator function at $x_{z'}(\theta)$. Rather than include an additional (weak) upper-bound, $(N+1)/2$, on the cost accrued because we must solve all N of the $V(N-i)$ problems explicitly (rather than as few as one additional problem), we leave this factor for a tighter empirical analysis. For example, if $\nu = \gamma = 1.2$, then with $c = \rho^{1/1-p}$, we have a slow down of 3.1, 3.7, and 11, for $p = 0.7, 0.8$ and 0.9 .

Conclusions

Anytime VCG mechanisms are proposed to provide anytime optimization with self-interested agents and retain strategyproofness whenever the mechanism is interrupted and asked for an answer. An anytime mechanism can solve easy instances optimally, while terminating early with approximate solutions on very hard instances. In doing so, we exposed an intriguing tension between flexibility and strategyproofness, through the analysis of agent-independent stage estimators.

In future work we intend to complete an experimental validation of the time-robustness approach, together with a calibration of the γ - and ν -assumptions in more realistic domains. We are interested to understand the cost of providing robustness against attempts to manipulate via run time, and the sensitivity to the probability p . In addition, there should be plenty of opportunity to explore smart methods to expand the range across time and to avoid duplication of effort.

Acknowledgments

This work is supported in part by NSF grant IIS-0238147, and by the Harvard College Research Program.

References

- Andersson, A.; Tenhunen, M.; and Ygge, F. 2000. Integer programming for auctions with bids for combinations. In *Proc. 4th International Conference on Multi-Agent Systems (ICMAS-00)*.
- Archer, A.; Papadimitriou, C.; Talwar, K.; and Tardos, E. 2003. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, 205–214.
- Banks, J. S.; Ledyard, J. O.; and Porter, D. 1989. Allocating uncertain and unresponsive resources: An experimental approach. *The Rand Journal of Economics* 20:1–25.
- Bartal, Y.; Gonen, R.; and Nisan, N. 2003. Incentive compatible multi unit combinatorial auctions. Technical report, The Hebrew University of Jerusalem.
- de Vries, S., and Vohra, R. V. 2003. Combinatorial auctions: A survey. *Inform Journal on Computing* 15(3):284–309.
- Dean, T. L., and Boddy, M. 1988. An analysis of time-dependent planning. In *Proc. 7th National Conference on Artificial Intelligence (AAAI-88)*, 49–54.
- Fischetti, M., and Lodi, A. 2002. Local branching. Technical Report OR-02-6, University of Bologna.
- Goldberg, A., and Hartline, J. 2003. Competitiveness via consensus. In *Proc. 14th ACM Symposium on Discrete Algorithms*.
- Holzman, R.; Kfir-Dahav, N.; Monderer, D.; and Tennenholtz, M. 2001. Bundling equilibrium in combinatorial auctions. *Games and Economic Behavior*.
- Horvitz, E. J. 1987. Reasoning about beliefs and actions under computational resource constraints. In *Proc. 3rd AAAI Workshop on Uncertainty in Artificial Intelligence*, 429–444.
- Lavi, R.; Mu'alem, A.; and Nisan, N. 2003. Towards a characterization of truthful combinatorial auctions. In *Proc. 44th Annual Symposium on Foundations of Computer Science*.
- Lehmann, D.; O'Callaghan, L. I.; and Shoham, Y. 2002. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM* 49(5):577–602.
- Leyton-Brown, K.; Nudelman, E.; and Shoham, Y. 2002. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In *Proc. 8th Int. Conf. on Principles and Practice of Constraint Programming*.
- Mu'alem, A., and Nisan, N. 2002. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proc. 18th National Conference on Artificial Intelligence (AAAI-02)*.
- Nisan, N., and Ronen, A. 2000. Computationally feasible VCG mechanisms. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 242–252.
- Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 1–12.
- Rothkopf, M. H.; Pekeč, A.; and Harstad, R. M. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.
- Sandholm, T. W.; Suri, S.; Gilpin, A.; and Levine, D. 2001. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. 17th Int. Joint Conf. on Artificial Intelligence*.
- Sandholm, T. W. 1999. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 542–547.